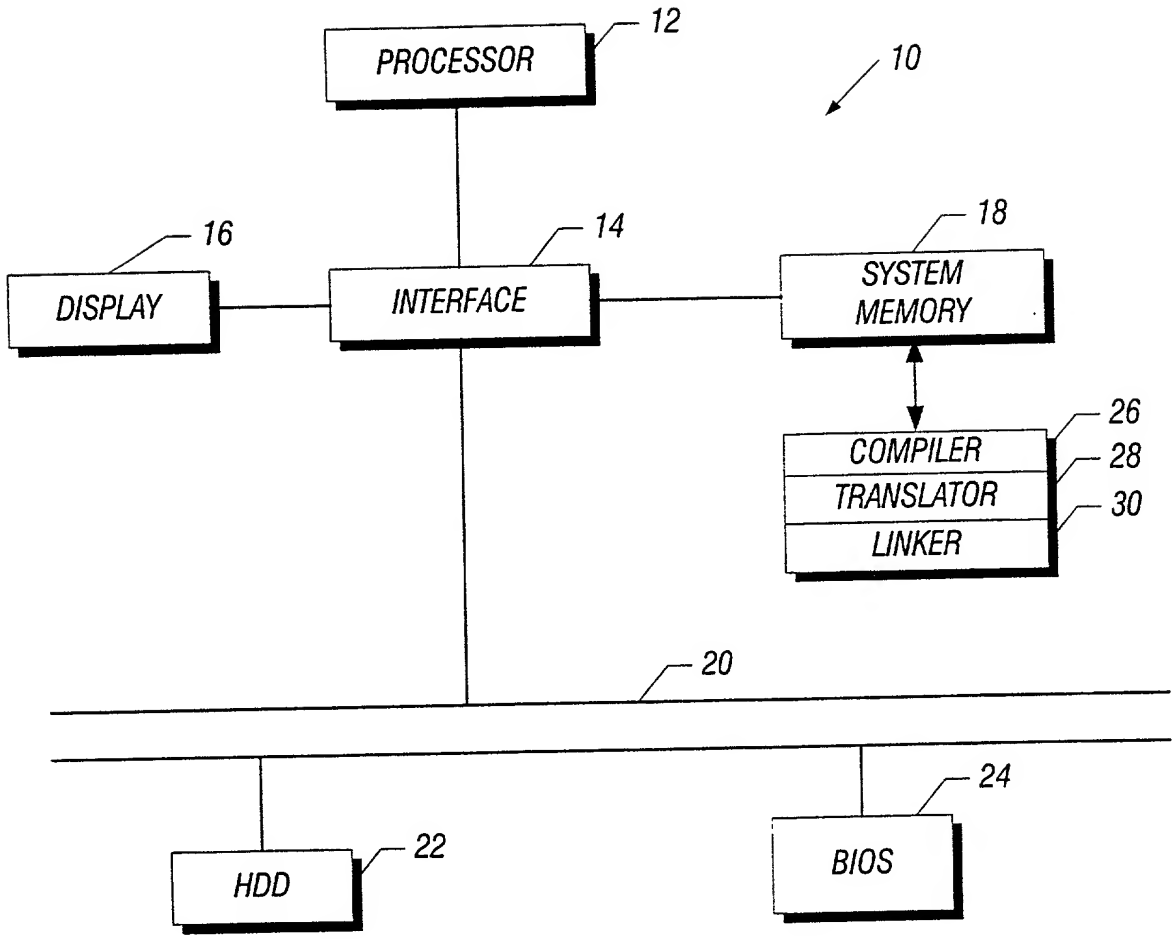
PROCESSOR
12
10
16
DISPLAY
14
INTERFACE
18
SYSTEM MEMORY
26
COMPILER
28
TRANSLATOR
30
LINKER
20
22
HDD
24
BIOS

FIG. 1

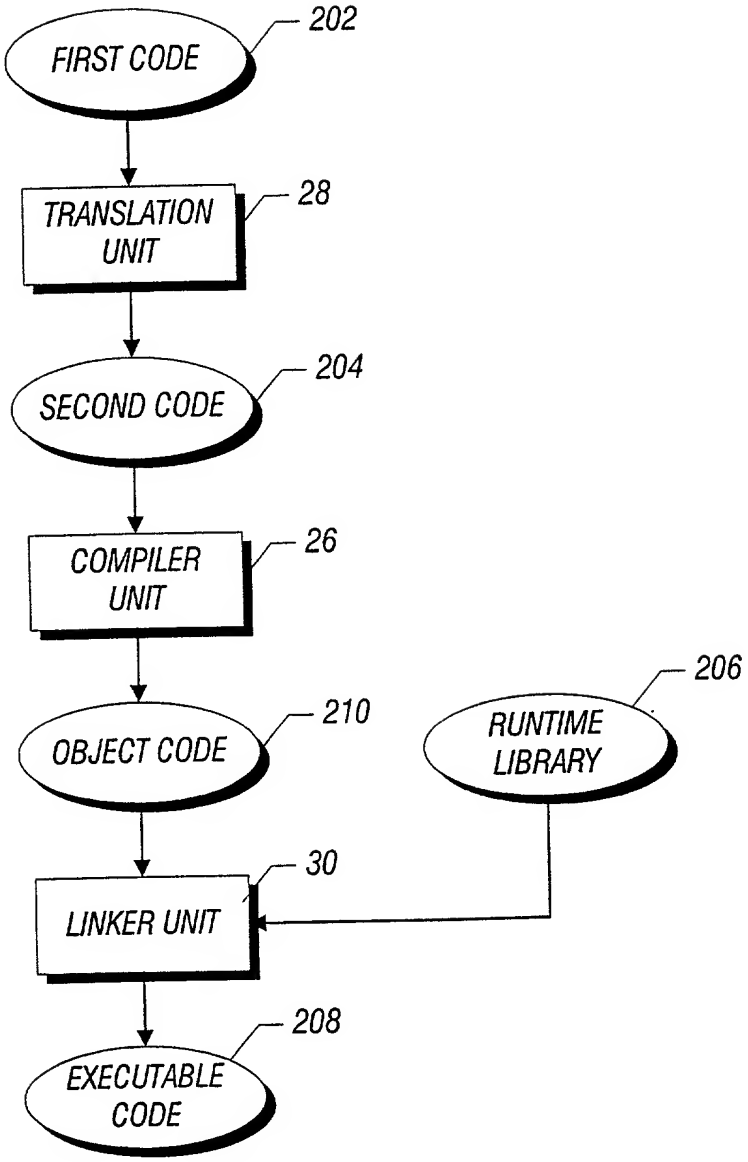
FIRST CODE
202
TRANSLATION UNIT
28
SECOND CODE
204
COMPILER UNIT
26
OBJECT CODE
210
RUNTIME LIBRARY
206
LINKER UNIT
30
EXECUTABLE CODE
208

FIG. 2

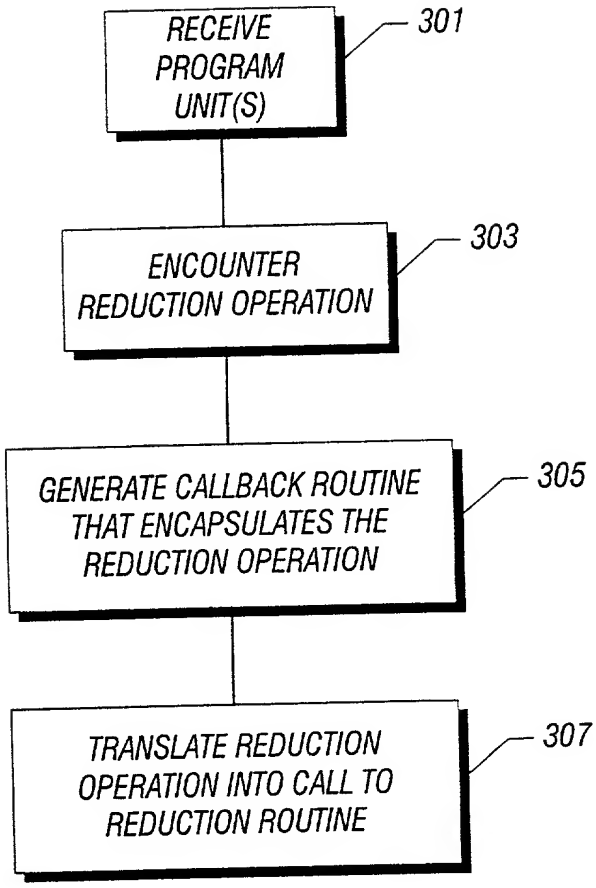
RECEIVE PROGRAM UNIT(S)
301
ENCOUNTER REDUCTION OPERATION
303
GENERATE CALLBACK ROUTINE THAT ENCAPSULATES THE REDUCTION OPERATION
305
TRANSLATE REDUCTION OPERATION INTO CALL TO REDUCTION ROUTINE
307

FIG. 3

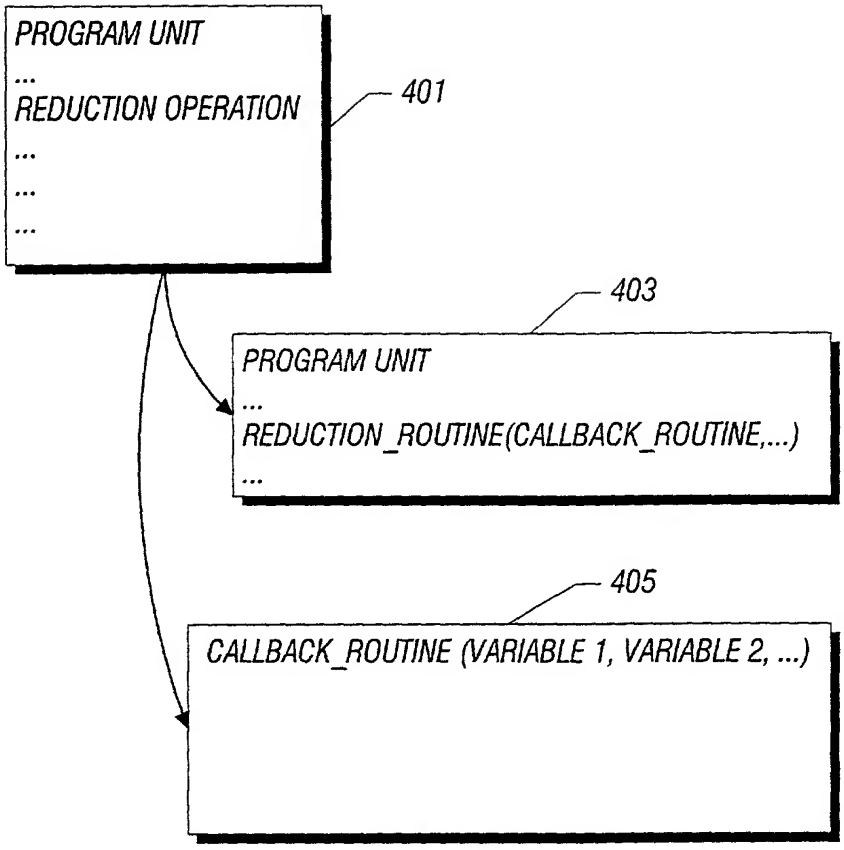
PROGRAM UNIT
...
REDUCTION OPERATION
...
...
...
401
403
PROGRAM UNIT
...
REDUCTION_ROUTINE(CALLBACK_ROUTINE,...)
...
405
CALLBACK_ROUTINE (VARIABLE 1, VARIABLE 2, ...)

FIG. 4

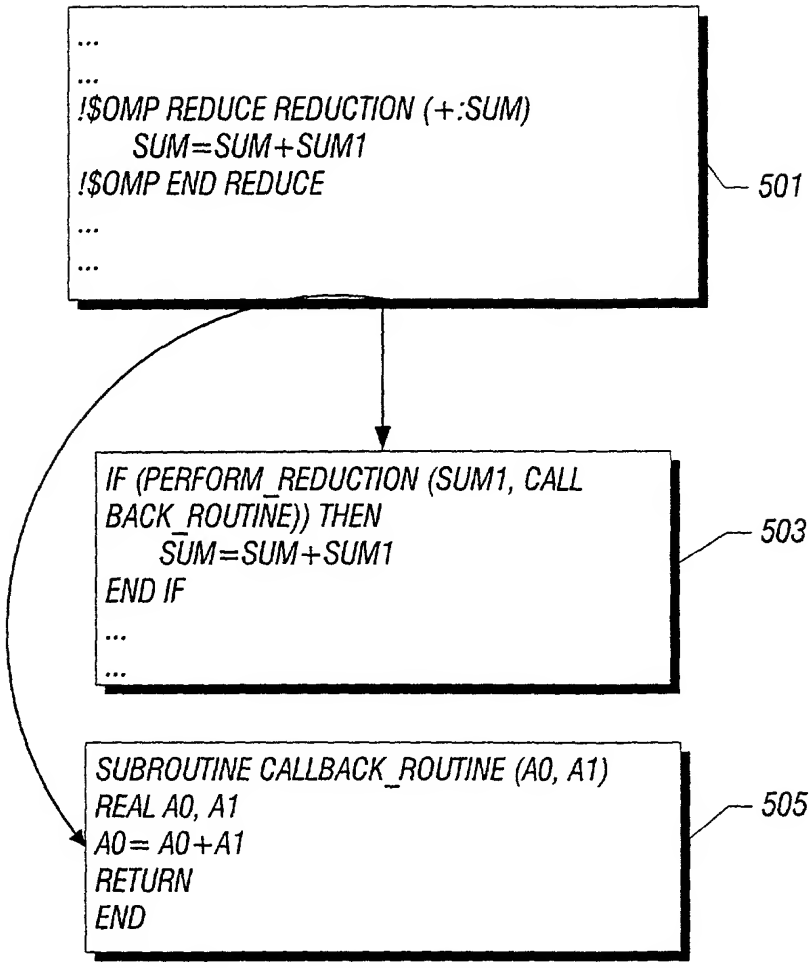
...
...
!$OMP REDUCE REDUCTION (+:SUM)
SUM=SUM+SUM1
!$OMP END REDUCE
...
...
501
IF (PERFORM_REDUCTION (SUM1, CALL
BACK_ROUTINE)) THEN
SUM=SUM+SUM1
END IF
...
...
503
SUBROUTINE CALLBACK_ROUTINE (A0, A1)
REAL A0, A1
A0= A0+A1
RETURN
END
505

FIG. 5

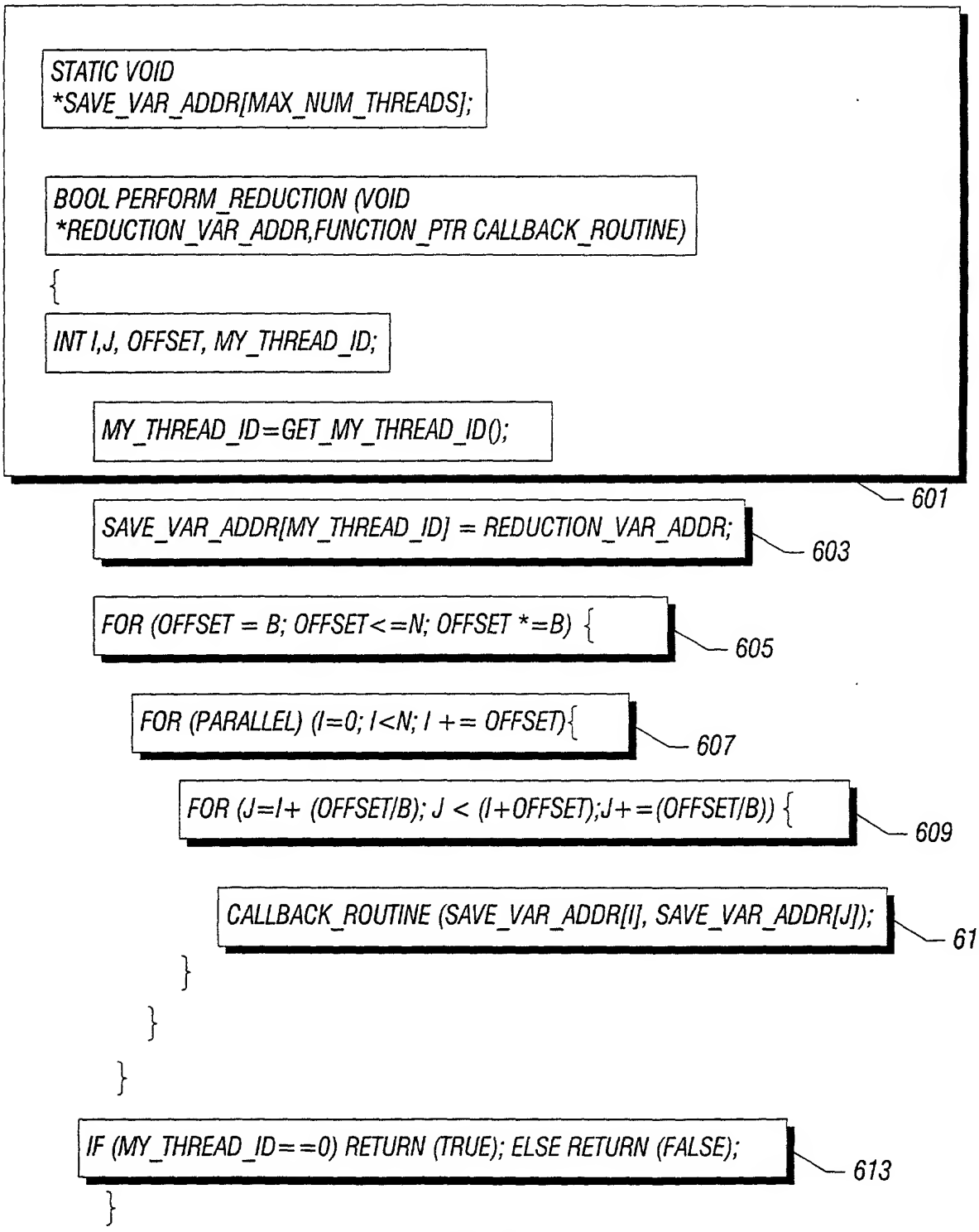
STATIC VOID
*SAVE_VAR_ADDR[MAX_NUM_THREADS];
BOOL PERFORM_REDUCTION (VOID
*REDUCTION_VAR_ADDR,FUNCTION_PTR CALLBACK_ROUTINE)
{
INT I,J, OFFSET, MY_THREAD_ID;
MY_THREAD_ID=GET_MY_THREAD_ID();
601
SAVE_VAR_ADDR[MY_THREAD_ID] = REDUCTION_VAR_ADDR;
603
FOR (OFFSET = B; OFFSET<=N; OFFSET *=B) {
605
FOR (PARALLEL) (I=0; I<N; I += OFFSET){
607
FOR (J=I+ (OFFSET/B); J < (I+OFFSET);J+=(OFFSET/B)) {
609
CALLBACK_ROUTINE (SAVE_VAR_ADDR[I], SAVE_VAR_ADDR[J]);
611
}
}
}
IF (MY_THREAD_ID==0) RETURN (TRUE); ELSE RETURN (FALSE);
613
}

FIG. 6

600

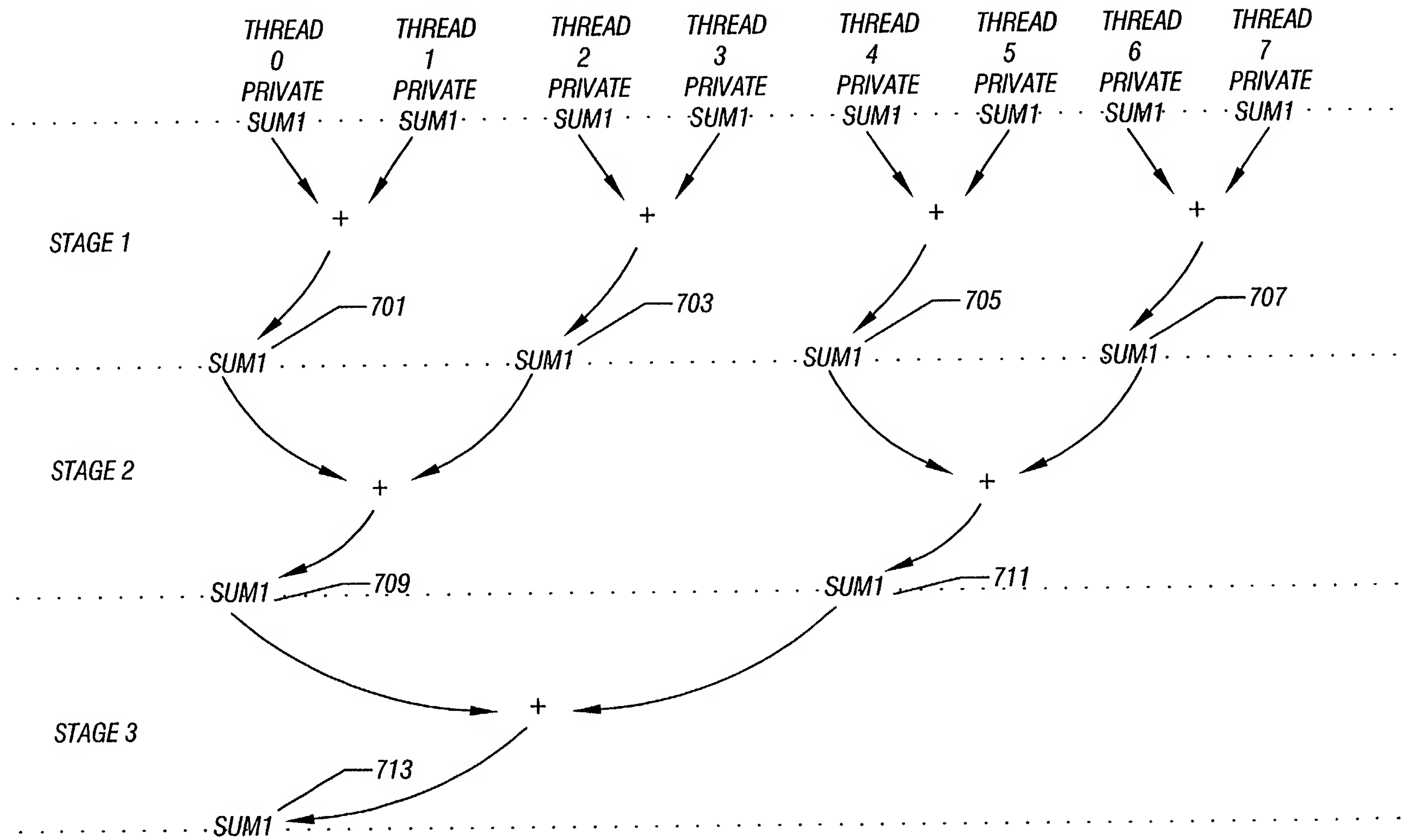
THREAD 0 PRIVATE SUM1
THREAD 1 PRIVATE SUM1
THREAD 2 PRIVATE SUM1
THREAD 3 PRIVATE SUM1
THREAD 4 PRIVATE SUM1
THREAD 5 PRIVATE SUM1
THREAD 6 PRIVATE SUM1
THREAD 7 PRIVATE SUM1
STAGE 1
STAGE 2
STAGE 3
SUM1
701
703
705
707
709
711
713

FIG. 7

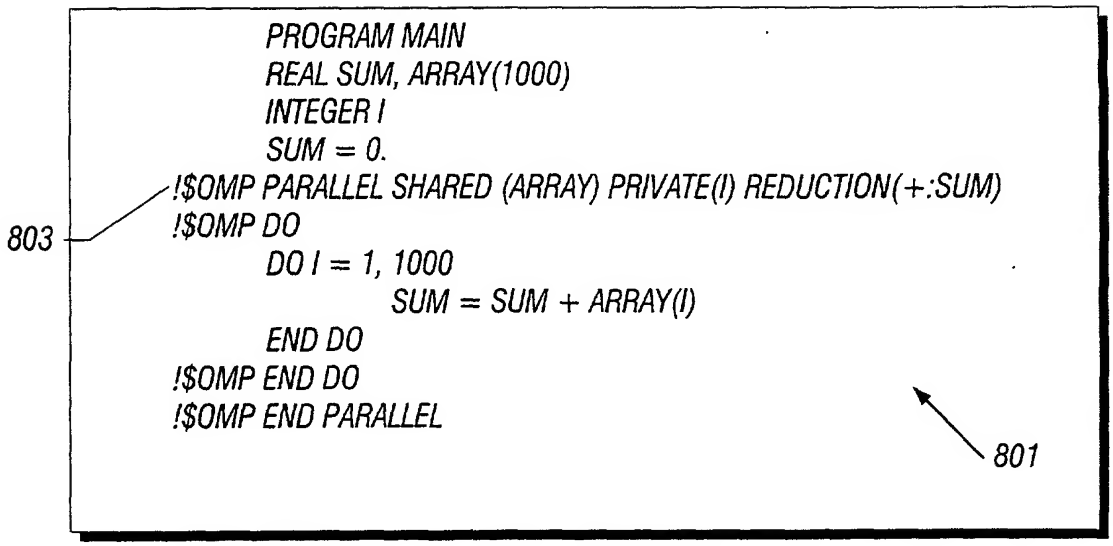
PROGRAM MAIN
REAL SUM, ARRAY(1000)
INTEGER I
SUM = 0.
!$OMP PARALLEL SHARED (ARRAY) PRIVATE(I) REDUCTION(+:SUM)
!$OMP DO
DO I = 1, 1000
SUM = SUM + ARRAY(I)
END DO
!$OMP END DO
!$OMP END PARALLEL
803
801

FIG. 8

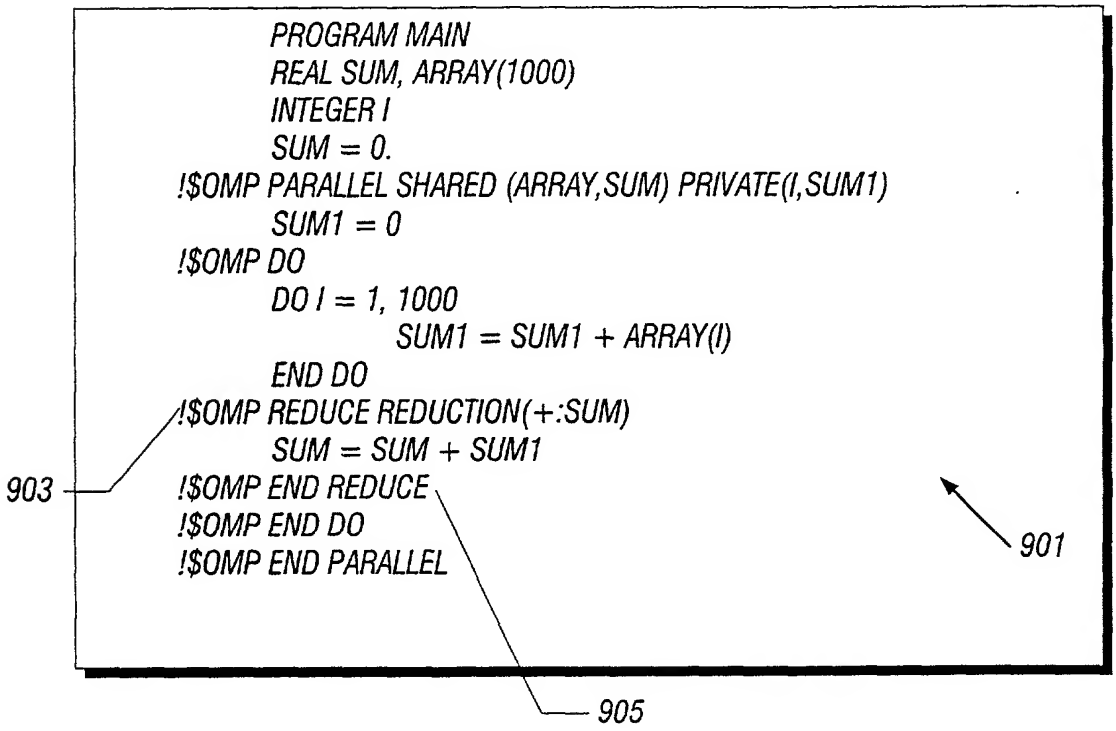
PROGRAM MAIN
REAL SUM, ARRAY(1000)
INTEGER I
SUM = 0.
!$OMP PARALLEL SHARED (ARRAY,SUM) PRIVATE(I,SUM1)
SUM1 = 0
!$OMP DO
DO I = 1, 1000
SUM1 = SUM1 + ARRAY(I)
END DO
!$OMP REDUCE REDUCTION(+:SUM)
SUM = SUM + SUM1
!$OMP END REDUCE
!$OMP END DO
!$OMP END PARALLEL
903
901
905

FIG. 9

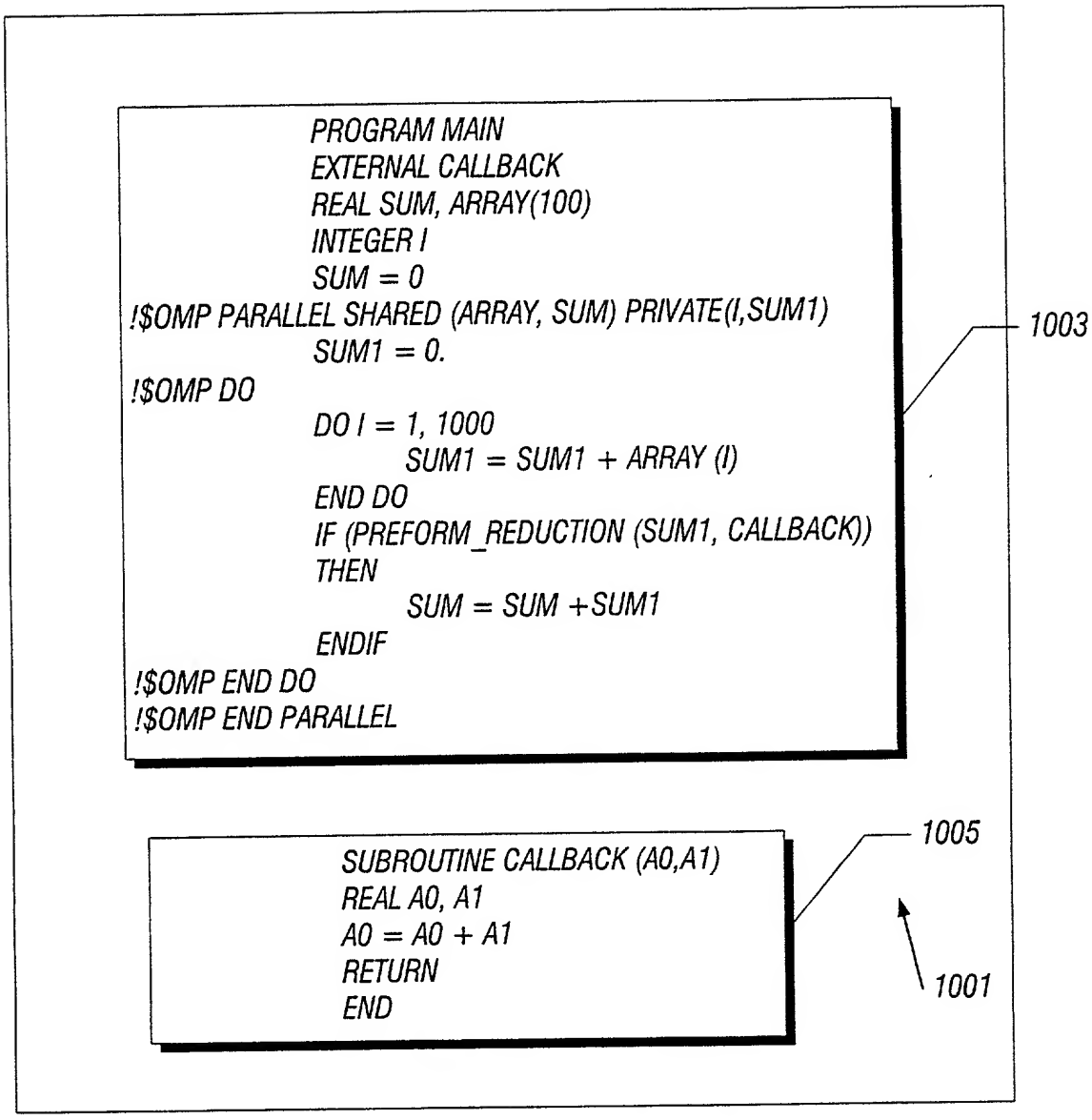
PROGRAM MAIN
EXTERNAL CALLBACK
REAL SUM, ARRAY(100)
INTEGER I
SUM = 0
!$OMP PARALLEL SHARED (ARRAY, SUM) PRIVATE(I,SUM1)
SUM1 = 0.
!$OMP DO
DO I = 1, 1000
SUM1 = SUM1 + ARRAY (I)
END DO
IF (PREFORM_REDUCTION (SUM1, CALLBACK))
THEN
SUM = SUM +SUM1
ENDIF
!$OMP END DO
!$OMP END PARALLEL
1003
SUBROUTINE CALLBACK (A0,A1)
REAL A0, A1
A0 = A0 + A1
RETURN
END
1005
1001

FIG. 10